

(19)

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 0 833 517 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
01.04.1998 Bulletin 1998/14

(51) Int Cl. 6: H04N 7/30

(21) Application number: 97307141.8

(22) Date of filing: 15.09.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC
NL PT SE

- Wus, John
Warminster, Pennsylvania (US)
- Schmidt, Robert Louis
Howell, New Jersey 07731 (US)
- Li, Weiping
Bethlehem, PA 18017 (US)

(30) Priority: 25.09.1996 US 27436 P
07.03.1997 US 813218

(71) Applicant: AT&T Corp.
New York, NY 10013-2412 (US)

(74) Representative: Pearce, Anthony Richmond
MARKS & CLERK,
Alpha Tower,
Suffolk Street Queensway
Birmingham B1 1TT (GB)

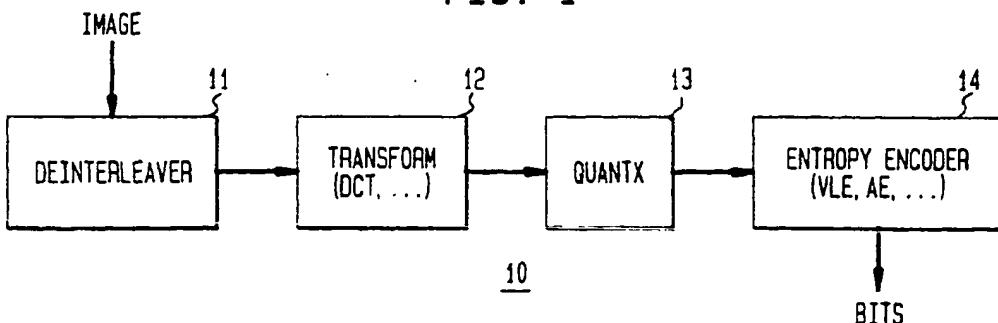
(72) Inventors:
• Puri, Atul
Riverdale, New York 10463 (US)

(54) Fixed or adaptive deinterleaved transform coding for image coding and intra coding of video

(57) A coding strategy efficiently codes intra (macroblocks, regions, pictures, VOP) data. This strategy uses two basis approaches, a Fixed Deinterleaved Transform Coding approach, and an Adaptive Deinterleaved Transform Coding approach. Furthermore, within each

approach, two types of coders are developed. One coder operates on an entire picture or VOPs and the other coder operates on small local regions. Using coders and decoders of the present invention, efficient coding at a range of complexities becomes possible, allowing suitable tradeoffs for a variety of applications.

FIG. 1



Description**BACKGROUND OF THE INVENTION**

This patent application is based on provisional application 60/027,436 filed on September 25, 1996.

The present invention relates generally to methods and apparatuses for coding images and intra coding of video and more particularly to a method and apparatus for coding images and intra coding of video using transform coding. Intra coding is important for applications requiring simple encoding/decoding, low delay, a high degree of error robustness or a high level of interactivity. Examples of such applications include image/video on the Internet, wireless video, networked video games, etc.

The state of the art standards in image coding and intra coding of video employ transform coding (e.g., Discrete Cosine Transformation, DCT), which involves partitioning an image into non-overlapping blocks of 8x8 size and coding in units of a 2x2 array of luminance (Y) blocks and a corresponding chrominance block of Cr and a block of Cb signal (together referred to as a macroblock). Improvements in performance of intra coding have been obtained by predicting the DC coefficient of DCT blocks by using previously reconstructed DC coefficients. Recently, further improvements have been obtained in MPEG-4 by predicting AC coefficients as well.

Over the past several years, many researchers have followed a different approach that uses wavelets instead of transform coding. These researchers have reported substantial improvements, at the expense of increased complexity. Recently, variations of wavelet coding that use subsampling prior to coding have emerged (some of which are currently being experimented with in MPEG-4), that provide an even higher performance by using advanced quantization techniques that take advantage of subsampling, however, the complexity of such schemes is extremely high.

The present invention is therefore directed to the problem of developing a method and apparatus for coding images and video that has a high coding efficiency yet relatively low coding complexity.

SUMMARY OF THE INVENTION

The present invention solves this problem by using a deinterleaving step prior to the transformation step in combination with a suitable quantization technique. Deinterleaving is a more flexible form of subsampling. The method and apparatus of the present invention thus achieve the coding efficiency of wavelet coding, at the expense of only a small improvement in complexity over traditional transform coding but at a significantly reduced complexity relative to wavelet coding.

The present invention includes two forms of deinterleaving in the encoding process. According to the present invention, improved intra coding is achieved by

fixed or adaptive deinterleaving, transform (e.g., DCT), quantization with extensions and entropy encoding. (e.g., Variable Length Encoding, VLE, or Arithmetic Encoding, AE). According to the present invention, there are

5 two main approaches to coding. The first uses fixed deinterleaving, and the second uses adaptive deinterleaving. It now becomes possible as a result of the present invention to use a simple transform coding method that achieves high coding efficiency using a fixed deinterleaving approach, or to use a slightly more complex method that produces somewhat better results based on an adaptive deinterleaving approach, and yet both of these approaches are less complex than wavelet coding and achieving the same or nearly the same coding efficiency.

10 According to advantageous implementations of the present invention, for each approach, two variations exist, one intended for separate motion/texture part of the MPEG-4 Verification Model (VM) (global transform) coding and the other intended for combined motion/texture part of the VM (local transform) coding.

BRIEF DESCRIPTION OF THE DRAWINGS

25 FIG 1 depicts the basic encoder block diagram for Global Deinterleaved Transform (GDT) coding according to the present invention.

FIG 2 shows the corresponding decoder for the encoder shown in FIG 1 according to the present invention.

30 FIG 3 depicts the basic encoder block diagram of Local Deinterleaved Transform (LDT) coding according to the present invention.

FIG 4 shows the corresponding decoder for the encoder shown in FIG 3 according to the present invention.

35 FIG 5 shows a simple example of deinterleaving a region by a factor of 2 both in the horizontal and vertical direction to generate subregions.

40 FIG 6 shows an 8x8 array of subpictures, each 22x18 in size that results from application of 8:1 deinterleaving in horizontal and vertical directions to the luminance signal in GDT coding for _____ (QCIF) resolution.

45 FIG 7 shows a 4x4 array of subpictures each 8x8 in size that result from application of 4:1 deinterleaving in horizontal and vertical directions to 32x32 regions of luminance signal in LDT coding for QCIF resolution.

FIG 8 depicts one method of extended quantization, QuantX Method 1, according to the present invention.

50 FIG 9 shows the inverse operation of the extended quantization method shown in FIG 8, QuantX Method 1, according to the present invention.

FIG depicts an example of quantized DC coefficients prediction used in the present invention.

55 FIG 11 shows an example of AC coefficient prediction structure employed in the present invention.

FIG 12 depicts another method of extended quantization, QuantX Method 2, according to the present invention.

FIG 13 shows the inverse operation of the extended quantization method shown in FIG 12, QuantX Method 2, according to the present invention.

FIG 14 depicts another method of extended quantization, QuantX Method 3, according to the present invention.

FIG 15 shows the inverse operation of the extended quantization method shown in FIG 14, QuantX Method 3, according to the present invention.

FIG 16 shows the block diagram of an Adaptive Global Deinterleaved Transform (AGDT) encoder 150 used in the present invention.

FIG 17 shows the block diagram of the AGDT decoder used in the present invention, which corresponds to the AGDT encoder shown in FIG 16.

FIG 18 shows a block diagram of an Adaptive Local Deinterleaved Transform (ALDT) encoder used in the present invention.

FIG 19 shows the ALDT decoder used in the present invention, which corresponds to the encoder of FIG 17.

FIG 20 shows an example of quadtree segmentation employed in the present invention.

DETAILED DESCRIPTION

According to the present invention, a significant improvement in efficiency of intra coding within the framework of transform coding is now possible. The present invention has been designed for use with MPEG-4. Often in MPEG-1/2 video coding, intra coding is employed for pictures coded by themselves, which are called I-pictures, or for intra macroblocks in predictively coded pictures (P-pictures or B-pictures). Besides pictures or macroblocks, MPEG-4 also introduces the concept of Video Objects (VO) and Video Object Planes (VOPs).

In MPEG-4 coding a scene can be partitioned into a number of Video Objects each of which can be coded independently. A VOP is a snapshot in time of a Video Object. In fact, a picture then becomes a special case of a VOP that is rectangular in shape.

MPEG-4 coding also includes coding of VOPs of different types, such as I-VOPs, P-VOPs, and B-VOPs, which are generalizations of I-pictures, P-pictures, and B-pictures, respectively. Thus, in addition to coding of I-pictures, and intra macroblocks the present invention can also be used for coding of I-VOPs, both rectangular and arbitrary in shape.

The main functionality addressed by the present invention is a coding efficiency of I-VOPs, although this approach is extendable to coding of P-VOPs and B-VOPs. An additional functionality that could be indirectly derived from the present invention is spatial scalability.

The present invention achieves a significant improvement in intra coding efficiency (by a factor of 1.5 or more) and can be obtained while still employing the DCT coding framework by requiring the addition of deinterleaving and extended quantization. In general, intra

coding efficiency can also be somewhat improved by further improving DC coefficient prediction, incorporating AC coefficient predictions and scanning adaptations, however, even after combining all these techniques the improvements may be relatively small as compared to the potential improvement of the present invention.

For I-VOPs coding in MPEG-4, FIG 1 depicts the basic encoder 10 block diagram of Global Deinterleaved Transform (GDT) coding according to the present invention. At the input to the deinterleaver 11, the image is in pixel format with each pixel being represented by the three components of luminance, chrominance and saturation, which are digital values. These digital values are fed into the deinterleaver 11, which separates contiguous samples within the image. In other words, the deinterleaver 11 separates the pixels sets into a number of pixel subsets, but does so by creating subsets from non-contiguous pixels. This then requires specification of the separation pattern used. Each subset contains several of the digital samples, however, the samples within a given subset were not contiguous with each other in the original image.

The transform operation 12 converts the digital pixel values within each subset into transform coefficients, such that most of the energy is packed in a few coefficients. In this step, for example, Discrete Cosine Transform (DCT) can be used. Other known transform techniques can also be used, such as _____.

The transformer 12 receives the pixel subsets from the deinterleaver 11. Each subset contains several pixels, with each pixel being represented by the three values of chrominance, luminance and saturation, or some equivalent color system. The transformer 12 then outputs coefficients representing the spatial frequency components of the values within each subset. While there is no real compression at this point, the DCT transform groups the data that enables the latter processes to significantly reduce the data. The DCT transform defines most of the information within the subset in the lower spatial frequencies and many of the higher spatial frequencies come out to be zero, resulting in compression later on. The output of the transformer 12 then is a block of coefficients, one block for each subset created by the deinterleaving process 11.

The QuantX process 13 includes normal quantization plus some extensions to improve coding efficiency and prepares the data for entropy encoding 14. This will be described in detail below, as three different QuantX processes 13 are presented herein. The output of the QuantX process 13 is a block of bits, one for each subset created in the deinterleaving process 11.

Following the QuantX process 13 is the encoding process 14. In this case entropy encoding is used. Any form of entropy encoding will suffice. Variable Length Encoding (VLE) is one example of entropy encoding. Arithmetic Encoding (AE) is another example. The coded bitstream generated by the Entropy Encoder 14 can

now be stored or transmitted.

Other known entropy encoding can be used.

FIG 2 shows the decoder 20 corresponding to the encoder 10 shown in FIG 1. The decoder 10 for Global Deinterleaved DCT Encoding of I-VOPs includes an entropy decoder 21, an inverse quantization 22, an inverse transform 23 and a reinterleaver 24. The entropy decoder 21 inverts codewords back to coefficient data. The inverse quantization 22 performs the inverse operation of the quantization 13 plus some extensions performed in the encoder 10. The inverse transform 23 performs the inverse operation of the transform 12, and the reinterleaver 24 performs the inverse of the deinterleaver 11.

The coded bitstream is fed into the entropy decoder 21. The entropy decoder 21 outputs a block of data similar to that input to the entropy encoder 14. Due to the deinterleaving 11 performed on the coding side, this block of data is subgrouped into blocks corresponding to the subsets created by the deinterleaver 11. The entropy decoder 21 outputs the block of data to the Inverse QuantX 22.

The Inverse QuantX 22 will be described in detail below, as three different processes are presented herein, depending upon the coding process performed. The Inverse QuantX 22 feeds its output to the inverse transform 23. The output of the Inverse QuantX 22 is a block of coefficients, which can be further subgrouped according to the subsets created by the deinterleaving process 11.

The inverse transform 23 then performs the inverse transform operation on each subblock of coefficients to convert it to a subblock of pixel subsets. These subblocks are now ready for reinterleaving. The reinterleaver 24 then reconstructs the original order in which the pixels appeared.

FIG 3 depicts the basic encoder 30 block diagram of LDT coding. The main difference with respect to FIG 1 is that prior to deinterleaving an input VOP or picture is segmented into local regions. These regions can be either square (blocks) or even arbitrary in shape. With such segmentation, some of the coding details related to transform size and QuantX may change.

The image in pixel format is fed into the local regions segmenter 31, which outputs the segmented image signal to the deinterleaver 32. In this case, the local regions segmenter 31 creates subsets of pixels, which are contiguous. Then, in the deinterleaving step 32, these subsets are further partitioned so that the resulting partitioned subsets each contain non-contiguous pixels.

As in FIG 1, the remaining process is the same. The deinterleaver 32 passes its output to the transformer 33, which in turn feeds the QuantX 34, which feeds the entropy encoder 35, which outputs the coded bitstream.

FIG 4 shows the corresponding decoder 40 for the encoder 30 shown in FIG 3. The main difference with respect to the GDT decoder shown in FIG 2 is the addition of the local regions assembler 45 (e.g., block unfor-

matter) at the end of the decoding process. The local regions assembler 45 performs the inverse operation of the local regions segmenter 31. The LDT decoder 40 includes the entropy decoder 41, the inverse quantization 42, the inverse transform 43 the reinterleaver 44, and the local regions assembler 45.

The process in FIG 4 is identical to the process in FIG 2, except that each step in the process is being performed on a partitioned subset relative to FIG 2. For example, the encoded bits are fed into the entropy decoder 41. These bits are ordered by the local regions created in the encoding process 30. Thus, each step is performed on each local region group separately. The decoder 41 then outputs groups of blocks of bits to the QuantX 42, which then creates the groups of blocks of coefficients necessary for the inverse transform process 43. The inverse transform process 43 then outputs the groups of pixels to the reinterleaver 44, which reinterleaves the pixels within each group. Output from the reinterleaver 44 is therefore the local regions created by the local region segmenter. One can view the decoding process 40 in this case as being performed on each local region separately. At the end of this decoding process 40, the local regions are then assembled by the local regions assembler 45 to reconstruct the pixel image as it was presented to the local regions segmented 31.

We shall now describe some of the steps within the process in more detail. These include deinterleaving, and QuantX.

Deinterleaving

Deinterleaving is the process of separating an input picture (or a region) into subpictures (or subregions) such that the neighboring samples in the input picture (or region) are assigned to different subpictures (or subregions). The resulting subregions or subpictures thus contain samples that were not contiguous in the original picture.

FIG 5 shows a simple example of deinterleaving a region by a factor of 2 both in the horizontal and vertical direction to generate subregions. The original picture 51 being composed of pixels (o, x, +, -) is deinterleaved into four subpictures 52-55. Every other element of the first row (o, x, o, x, o, x) is then assigned to the first row of subpicture 51 (o, o, o) and the first row of the second subpicture 52 (x, x, x). The same is true for the remaining odd rows. The even rows are assigned to the third and fourth subpictures (+, +, +) and (-, -, -), respectively, and split as before. Essentially, each pixel (p_{ij}) is assigned to the subpicture_{k,m} where $k=\text{mod}(i/n)$ and $m=\text{mod}(j/n)$ and becomes element $p_{r,s}$ in that subpicture, where $r=(i-k)/n$ and $s=(j-m)/n$.

For example, if we let where $n=2$, as in FIG 5, we note that element 56 (i.e., p_{23}), is assigned to subpicture 01 (element 53), that is, $k=\text{mod}(2/2)=0$ and $m=\text{mod}(3/2)=1$. If we examine subpicture 01 (element 53), we note that element 57 appears as pixel 11 in that subpicture,

and $r=(i-k)/n=(2-0)/2=1$ and $s=(j-m)/n=(3-1)/2=1$.

In this example, in GDT coding, the deinterleaving factor is fixed to 8:1 for QCIF input resolution (176x144). For this resolution, FIG 6 shows an 8x8 array, 63, of subpictures, each subpicture 62 being 22x18 in size that results from application of 8:1 deinterleaving in horizontal and vertical directions to the luminance signal. Also, each of the chrominance components are deinterleaved by a factor of 4:1 resulting in a 4x4 array of subpictures, each of size 22x18.

On the other hand, in LDT coding, the deinterleaving factor is fixed to 4:1 for QCIF input resolution. FIG 7 shows a 4x4 array, 73, of subpictures each subpicture 72 being 8x8 in size that results from application of 4:1 deinterleaving in horizontal and vertical directions to 32x32 regions of luminance signal. In this case, the chrominance components are deinterleaved by a factor of 2:1 resulting in a 2x2 array of subregions, each 8x8 in size.

DCT

Two dimensional DCT is applied to deinterleaved subpictures or subregions. In GDT coding at QCIF resolution, the size of DCT is chosen to be 22x18 both for the luminance as well as the chrominance components. In LDT coding at QCIF resolution, the size of DCT is chosen to be 8x8 both for the luminance as well as the chrominance components.

QuantX Choices

The normal scalar quantization needs to be modified to take into account the fact that the transform coding is performed on deinterleaved data. Beyond quantization, the coefficient prediction of experiment may also be more effective in increasing coding efficiency due to higher correlation between coefficients of deinterleaved adjacent subpictures (subregions). Another approach is to exploit this correlation by forming vectors of coefficients of the same spectral frequency and performing DCT coding on such vectors (blocks). Finally, yet another alternative is to use vector quantization or a specific variation called Lattice Vector Quantization (LVQ) being examined in MPEG-4. These various approaches are referred to here as QuantX and offer different tradeoffs in performance versus complexity and the right one may be selected based on the application.

QuantX Method 1: Quantization and DCT Coefficient Predictions

This method is explained by reference to FIG 8. The signal input to the QuantX 80 is received by a quantizer 81, whose output signal is split. One path feeds a DC & AC Coefficient Predictor 82, and another path feeds one input of a subtractor 83. The output of the DC & AC Coefficient Predictor 82 feeds the other input of the sub-

tractor 83. The output of the DC & AC Coefficient Predictor 82 is subtracted from the output of the quantizer 81 and fed into a scanner 84, such as a zigzag scanner.

In GDT coding, the DCT coefficient subpictures of size 22x18 are quantized by the normal scalar quantization and then the coefficient subpictures are predicted based on previously quantized coefficient subpictures and coefficient difference subpictures are formed. In LDT coding, a very similar operation takes place on DCT coefficient subregions of size 8x8. The difference coefficients are scanned (e.g., zigzag scanned) to form (run, level) events.

FIG 9 shows the inverse operation of the QuantX shown in FIG 8. The signal input to the Inverse QuantX 90 is received by the Inverse Scanner 91. The output of the inverse scanner 91 is fed to one input of an adder 92. The second input of the adder 92 is from the output of a DC & AC Coefficient Predictor 93, which receives its input from the output of the adder 92. The output of the adder 92 is also passed to the inverse quantizer 94, which outputs the desired signal.

A scheme for quantized DC coefficients prediction is illustrated in FIG 10. In GDT coding, the DC prediction of leftmost subregion (subpicture) is selected to 128 while in LDT coding, using 1 bit overhead, selection between DC values of horizontally or vertically adjacent subregions (subpictures) is made. For the remaining subregions (subpictures) in the first row, DC prediction uses previous subregion (subpicture) DC value. For the first subregion (subpicture) of the second row, DC prediction is taken from the subregion (subpicture) above; all the other subregions (subpictures) of that row use Graham's predictor adaptively by selecting from horizontal and vertical adjacent subregion (subpicture) DC values without overhead. The prediction process for the second row is repeated for subsequent rows.

We now discuss how AC coefficient predictions are made. FIG 11 shows an example of AC coefficient prediction structure employed. In the case of LDT coding with 8x8 subregions, 2 rows and 2 columns of AC coefficient predictions for a subregion may be used. In case of larger subregion sizes in LDT coding or larger pictures in GDT coding, more AC coefficients may be predicted; the number and the structure of coefficients being predicted can be different but the basic principle of prediction remains the same.

For the left-most subregion (subpicture) of a region (picture), AC coefficient predictions are reset to 0. For the subsequent subregions (subpictures) in the first row of subregions, the L-shaped highlighted area (without DC coefficient) is predicted from subregions (subpicture). For the first subregion of second row of subregions, the same L-shaped area is predicted from the subregion immediately above. Subsequent subregions of second row are predicted by using first two columns of coefficients from previous subregion and the first two rows from the subregion above. There is an overlap of 1 coefficient (AC 11), which is resolved by averaging the

two prediction choices for this coefficient to generate a single prediction coefficient. The prediction process for the second row is repeated for subsequent rows.

Further, the potential of making the prediction process adaptive (with or without overhead) is also possible.

The difference coefficient subpictures of size 22x18 in GDT coding and subblocks of size 8x8 in LDT coding are zigzag scanned to form (run, level) events.

QuantX Method 2: Quantization and DCT of DCT Coefficient Vectors

FIG 12 illustrates the operations employed in this method of QuantX. In GDT coding of QCIF pictures, the DCT coefficient subpictures of size 22x18 are prequantized by a small quantization level ($Qp=2$ or 3) to reduce their dynamic range and then vectors (of 8x8 size for luminance and 4x4 size for chrominance) are generated by collecting all coefficients of the same frequency through all of the subpictures, DCT and quantized. In LDT coding with regions of size 32x32, a very similar operation takes place resulting in coefficient vectors (of 4x4 size for luminance and 2x2 size for chrominance); these vectors are DCT and quantized. In GDT coding, quantized DCT coefficient vectors of 8x8 size for luminance and 4x4 size for chrominance are zigzag scanned to form (run, level) events. In LDT coding, quantized DCT coefficient vectors of 4x4 size for luminance and 2x2 size for chrominance are zigzag scanned for form (run, level) events.

Referring to FIG 12, the QuantX 120 includes a prequantizer 121, a vector formatter 122, a transform 123, a quantizer 124 and a scanner 125. The prequantizer 121 receives the signal input to the QuantX 120, and outputs its signal to the vector formatter 122. The vector formatter feeds its output to the transform 123, which in turn feeds the quantizer 124, which feeds the scanner 125. The scanner outputs its signal as the output of the QuantX 120.

The Inverse Operation 130 of the QuantX 120 shown in FIG 12 is shown in FIG 13. The input to the inverse QuantX 130 is fed to the inverse scan 131, which feeds the inverse quantizer 132, which in turn feeds the inverse transform 133. The vector unformatter 134 receives the output from the inverse transform 133 and outputs its signal to the inverse prequantizer 135, whose output represents the output of the inverse QuantX 130.

QuantX Method 3: Lattice Vector Quantization of DCT Coefficient Vectors

FIG 14 illustrates the operations employed by this method of QuantX. The signal input to the QuantX 140 is received by the Vector Formatter 141, which passes its output to the Dimension Reducer 142, which in turn feeds its output to the Vector Quantizer 143. The Vector Quantizer 143 then passes its output to the Vector Quantization Indices Orderer 144, whose output repre-

sents the output of the QuantX 140.

In GDT coding of QCIF pictures, using DCT coefficient subpictures of size 22x18, vectors (of 8x8 size for luminance and 4x4 size for chrominance) are generated by collecting all coefficients of the same frequency through all of the subpictures; these vectors are quantized by the LVQ. In LDT coding of region size 32x32, a very similar operation takes place resulting in coefficient vectors (of 4x4 size for luminance and 2x2 size for chrominance); these vectors are also quantized by LVQ. Since VQ often requires small blocks for manageable size codebooks (or in LVQ, a manageable complexity), a reduction of vector dimension may be necessary and is accomplished in Dimension Reducer, which can be as simple an operation as dividing a vector of coefficients into sub-vectors or something more sophisticated. The process of LVQ is not described herein and is discussed in literature. Briefly, however, first LVQ of dimension 16 is tried, if it produces errors higher than a threshold then the LVQ of dimension 4 is tried. Also after LVQ, the LVQ indices of the entire picture or region may be ordered for increased efficiency, this process takes place in VQ Indices Orderer.

The Inverse Operation 150 of the QuantX 140 shown in FIG 14 is shown in FIG 15. The input to the inverse QuantX 150 is fed to the Vector Quantization Indices Reorderer 151, which feeds the inverse vector quantizer 152, which in turn feeds the dimension normalizer 153. The vector unformatter 154 receives the output from the dimension normalizer 153 and outputs its signal as the output of the inverse QuantX 150. As in the QuantX 140, in the Inverse QuantX 150, first LVQ of 16 is tried. If it produces errors higher than a threshold, then the LVQ of dimension 4 is tried. The specification of LVQ is the same as that in experiment T. 5 in the MPEG 4 trials.

Entropy Coding

We now discuss a VL coding and decoding method for coefficient (run, level) events, which are coded exploiting statistical variations to achieve even further efficiency.

In GDT coding if extended QuantX Method 1 is employed, a maximum run of 396 is possible and a level of at least ± 255 needs to be supported. For coding of luminance run/level events, the intra VLC table of U.S. Patent Application No. 08/####,###, entitled "Adaptive and Predictive Coding for Image Coding and Intra Coding of Video" by Puri, Schmidt and Haskell is employed. U.S. Patent Application No. 08/####,### is hereby incorporated by reference as if recited herein in its entirety. However, since this table supports only a maximum run of 64 and level of ± 128 (same as the MPEG-4 VM) it is extended to outside of this region by appending one extra bit for level and three extra bits for run and thus uses up to 25 bits. For coding of chrominance run/level events, the VLC table used is the one in the VM extend-

ed to support a maximum run of 396 and level of ± 255 by escaping outside of the currently supported region by appending one extra bit for level and three extra bits for run, thus using up to 26 bits. In case of LDT coding, since the subregion size is 8x8, the VLC table of the aforementioned earlier patent application that was incorporated by reference, is employed for luminance and the VLC table of VM is employed for chrominance; both of these tables do not require any extensions.

If extended quantization QuantX Method 2 is employed, in GDT coding, since the vector size is 8x8, the VLC table of the previously incorporated by reference patent application is employed for luminance and the VLC table of VM is employed for chrominance; both of these tables do not require any extensions. In the case of LDT coding, a maximum run for luminance is 15 and that for chrominance is 3; in this case new tables which are subsets of the previously incorporated patent application are employed.

If extended quantization QuantX Method 3 is employed, VLC tables used are based on tables available in MPEG-4 core experiment T5 and are available publicly.

Adaptive Deinterleaved Transform Coding

Further improvements in DT coding are possible by using an encoding structure as shown in FIG 16, which shows the block diagram of an Adaptive Global Deinterleaved Transform (AGDT) encoder 160. The major difference with respect to FIG 1 is that a quadtree segmentation is employed by using a Quadtree Segmenter on an entire picture or VOP basis prior to the deinterleaver and is adaptive rather than fixed segmentation. Thus, the deinterleaving is only performed on only the portions identified by Global Quadtree Segmenter to be worth deinterleaving while others are coded without deinterleaving. The operation of other blocks is similar to that discussed for fixed GDT.

Referring to FIG 16, the image is fed into the Global Quadtree Segmenter 161, whose output is passed to the deinterleaver 162, which in turn passes its output to the transform 163. The QuantX 164 receives the output from the transform 163 and passes its output to the entropy encoder 165, which outputs the coded bitstream.

FIG 17 shows the block diagram of the AGDT decoder 170 corresponding to the AGDT encoder 160 shown in FIG 16. The coded bitstream is fed into the entropy decoder 171, the output of which is passed to the inverse QuantX 172, which in turn passes its output to the inverse transform 173. The reinterleaver 174 receives the output from the inverse transform 173 and feeds its output to the Global Quadtree Assembler 175, which outputs the reconstructed image.

FIG 18 shows a block diagram of an Adaptive Local Deinterleaved Transform (ALDT) encoder 180. The major difference with respect to FIG 16 is that the quadtree segmentation is applied locally (on regions) rather than

the entire picture or VOP. Deinterleaving is then performed on the region identified by the Local Quadtree Segmenter as worth deinterleaving. The remaining blocks are similar to those described above.

5 The image signal is input to the Local Quadtree Segmenter 181, whose output is fed to the deinterleaver 182, which passes its output to the transform 183. The QuantX 184 receives the output from the transform 183 and passes its output to the entropy encoder 185, which outputs the coded bitstream.

10 FIG 19 shows the ALDT decoder 190 that corresponds to the encoder of FIG 18. The coded bits are fed into the entropy decoder 191, which passes its output to the inverse QuantX 192, which in turn passes its output to the inverse transform 193, which in turn passes its output to the reinterleaver 194. The local quadtree assembler 195 receives the output of the reinterleaver 194 and outputs the reconstructed image.

Quadtree Segmenter

As shown in FIGs 16 and 18, quadtree segmentation is employed prior to deinterleaving to allow adaptation of the amount of deinterleaving to the spatial content of the picture being coded.

15 An example of quadtree segmentation employed is shown in FIG 20; both GDT and LDT use this type of segmentation, the only difference being in the number of levels employed -- GDT employs $_$ levels of segmentation, whereas LDT employs $_$ levels of segmentation.

20 As shown in FIG 20, picture block 200 is segmented into subblocks 202-205. Subblock 203 is then further partitioned into sections 206-209. The remaining blocks were not segmented to indicate that this process only segments the necessary blocks.

Syntax and Semantics for MPEG-4

We now provide the necessary syntax and semantics 40 needed for generating coded bitstreams using the present invention. The various classes referred to below correspond to the current syntax of MPEG-4 VM3.2

VideoSession Class

45 No changes are necessary for this class.

VideoObject Class

50 No changes are necessary for this class.

VideoObject Layer Class

55 No changes are necessary for this class.

VideoObject Plane Class

Two new syntax elements are introduced in this

class as follows.

region_size
deinterleave_ratio

These syntax elements are defined as follows.

region_size

This is a 3 bit code which specifies the size of the region on which deinterleaving is performed prior to coding. The size of the region for each code is shown in Table 1 as follows:

Table 1

Code	Meaning
000	16x16
001	32x32
010	64x64
011	128x128
100	reserved
101	reserved
110	reserved
111	full picture

deinterleave_ratio

This is the 3-bit code which specifies the amount of deinterleaving performed on the identified region prior to coding. The same deinterleaving ration is used both horizontally and vertically. The amount of deinterleaving for each code is shown in Table 2 as follows.

Table 2

Code	Meaning
000	1:1
001	2:1
010	4:1
011	8:1
100	16:1
101	reserved
110	reserved
111	reserved

Region Class

Data for each region consists of region header followed by subregion data.

Table 3

Rtype	Rquant	Subregion data
-------	--------	----------------

5 Rquant

Rquant is a 3 bit quantizer that takes nonlinear values bounded by 1 to 31 with the meaning as shown in Table 4.

Table 4

Code	Qp
000	2
001	43
010	7
011	10
100	14
101	18
110	23
111	28

Subregion Class

The definition of subregion data is dependent on the QuantX method employed and is specified as follows:
For QuantX Method 1:

Table 5

Structure of Subregion Class for QuantX Method 1	
Cod_subreg	Tcoefs_subreg

Cod_subreg

40 Cod-subreg is a 1 bit flag that identifies if there is any coded data (nonzero values) for that subregion.

Tcoefs_subreg

45 Tcoefs_subreg are differential quantized coefficients of subregion.
For QuantX Method 2:

Table 6

Structure of Subregion Class for QuantX Method 2	
Cod_vector	Tcoefs_vector

Cod_vector

55 Cod-vector is a 1 bit flag that identifies if there is any coded data a subregion.

Tcoefs_vector

Tcoefs_vector refers to twice quantized coefficients of a vector.

For QuantX Method 3:

5

Claims

1. A method for coding an image comprising the steps of:

a) deinterleaving the image to form a plurality of image subsets;

b) transforming the plurality of image subsets into a plurality of transform coefficients;

c) converting the plurality of transform coefficients to a plurality of samples; and

d) performing an entropy encoding of the plurality of samples to form an encoded bit stream.

15

20

2. A method for decoding a bit stream representing an image that has been encoded, comprising the steps of:

a) performing an entropy decoding of the bit stream to form a plurality of samples;

b) converting the plurality of samples to a plurality of transform coefficients;

c) performing an inverse transformation on the plurality of transform coefficients to form a plurality of image subsets; and

d) reinterleaving the plurality of image subsets to form the image.

25

30

3. The method according to claim 2, wherein the step b) of converting comprises the steps of:

(i) performing an inverse scanning of the plurality of samples to form a plurality of difference values;

(ii) adding the plurality of difference values to a plurality of predicted values to form a plurality of quantized values;

(iii) performing a DC and AC coefficient prediction on the plurality of quantized values to form the plurality of predicted values; and

(iv) performing an inverse quantization of the plurality of quantized values to form the plurality of transform coefficients.

40

45

4. The method according to claim 2, wherein the step b) of converting comprises the steps of:

(i) performing an inverse scan of the plurality of samples to form a plurality of quantized values;

(ii) performing an inverse quantization of the plurality of quantized values to form a plurality

50

55

of transformed vectors using a first quantization level;

(iii) performing an inverse transformation of the plurality of transformed vectors to form a plurality of vectors;

(iv) generating a plurality of quantized transform coefficients from the plurality of vectors; and

(v) performing an inverse quantization of the plurality of quantized transform coefficients to form the plurality of transform coefficients using a second quantization level, wherein said first quantization level is greater than said second quantization level.

5. The method according to claim 2, wherein the step b) of converting comprises the steps of:

(i) performing a vector quantization indices reordering of the plurality of samples to form a plurality of quantized vectors;

(ii) performing an inverse vector quantization of the plurality of quantized vectors to form a plurality of limited dimension vectors;

(iii) normalizing a dimension of the plurality of limited dimension vectors to form a plurality of vectors; and

(iv) unformatting the plurality of vectors to form the plurality of transform coefficients.

6. The method according to claim 2, wherein the step d) of reinterleaving further comprises the steps of:

(i) reinterleaving the plurality of image subsets to form a plurality of segments; and

(ii) assembling the plurality of segments formed by the reinterleaving step d) (i) into the image.

7. A method for coding an image comprising the steps of:

a) segmenting the image into a plurality of local regions;

b) deinterleaving the plurality of local regions to form a plurality of deinterleaved regions;

c) transforming the plurality of deinterleaved regions into a plurality of transform coefficients;

d) converting the plurality of transform coefficients to a plurality of samples; and

e) performing an entropy encoding of the plurality of samples to form an encoded bit stream.

8. A method for decoding a bit stream representing an image that has been encoded, comprising the steps of:

a) performing an entropy decoding of the bit stream to form a plurality of samples;

b) converting the plurality of samples to a plurality of transform coefficients;
 c) performing an inverse transformation of the plurality of transform coefficients to form a plurality of deinterleaved regions;
 d) reinterleaving the plurality of deinterleaved regions to form a plurality of local regions; and
 e) assembling the plurality of local regions to form the image.

9. The method according to claim 8, wherein the step b) of converting comprises the steps of:

- (i) performing an inverse scanning of the plurality of samples to form a plurality of difference values;
- (ii) adding the plurality of difference values to a plurality of predicted values to form a plurality of quantized values;
- (iii) performing a DC and AC coefficient prediction on the plurality of quantized values to form the plurality of predicted values; and
- (iv) performing an inverse quantization of the plurality of quantized values to form the plurality of transform coefficients.

10. The method according to claim 8, wherein the step b) of converting comprises the steps of:

- (i) performing an inverse scan of the plurality of samples to form a plurality of quantized values;
- (ii) performing an inverse quantization of the plurality of quantized values to form a plurality of transformed vectors using a first quantization level;
- (iii) performing an inverse transformation of the plurality of transformed vectors to form a plurality of vectors;
- (iv) generating a plurality of quantized transform coefficients from the plurality of vectors; and
- (v) performing an inverse quantization of the plurality of quantized transform coefficients to form the plurality of transform coefficients using a second quantization level, wherein said first quantization level is greater than said second quantization level.

11. The method according to claim 8, wherein the step b) of converting comprises the steps of:

- (i) performing a vector quantization indices reordering of the plurality of samples to form a plurality of quantized vectors;
- (ii) performing an inverse vector quantization of the plurality of quantized vectors to form a plurality of limited dimension vectors;
- (iii) normalizing a dimension of the plurality of limited dimension vectors to form a plurality of vectors; and
- (iv) unformatting the plurality of vectors to form the plurality of transform coefficients.

12. The method according to claim 8, wherein the step d) of reinterleaving further comprises the steps of:

- (i) reinterleaving the plurality of deinterleaved regions to form a plurality of local region segments; and
- (ii) assembling the plurality of local region segments to form the plurality of local regions.

13. An apparatus for coding an image comprising:

- a) a deinterleaver receiving the image and deinterleaving the image to form a plurality of image subsets;
- b) a transformer being coupled to the deinterleaver and transforming the plurality of image subsets into a plurality of transform coefficients;
- c) a converter being coupled to the transformer and converting the plurality of transform coefficients to a plurality of samples; and
- d) an encoder being coupled to the converter and performing an entropy encoding of the plurality of samples to form an encoded bit stream.

14. An apparatus for decoding a bit stream representing an image that has been encoded, comprising:

- a) a decoder receiving the bit stream and performing an entropy decoding of the bit stream to form a plurality of samples;
- b) a converter being coupled to the decoder and converting the plurality of samples to a plurality of transform coefficients;
- c) a reverse transformer being coupled to the converter and performing an inverse transformation of the plurality of transform coefficients to form a plurality of image subsets; and
- d) a reinterleaver being coupled to the reverse transformer and reinterleaving the plurality of image subsets to form the image.

15. The apparatus according to claim 14, wherein the converter comprises an inverse extended quantizer, said inverse extended quantizer including:

- a) an inverse scanner being coupled to the decoder and performing an inverse scan on the plurality of samples to form a plurality of difference values;
- b) an adder having a first input being coupled to the inverse scanner, having a second input, and having an output outputting a plurality of quantized values;

c) a coefficient predictor having an input being coupled to the output of the adder, performing a DC and AC coefficient prediction on the plurality of quantized values to form a plurality of predicted values, and having an output being coupled to the second input of the adder, wherein the plurality of quantized values equals a sum of the plurality of difference values and the plurality of predicted values; and

d) an inverse quantizer being coupled to the adder and inverse quantizing the plurality of quantized values to form the plurality of transform coefficients.

16. The apparatus according to claim 14, wherein the converter comprises an inverse extended quantizer, said inverse extended-quantizer including:

a) an inverse scanner being coupled to the decoder and performing an inverse scan of the plurality of samples to form a plurality of quantized values;

b) a first inverse quantizer being coupled to the inverse scanner and performing an inverse quantization of the plurality of quantized values to form a plurality of transformed vectors using a first quantization level;

c) an inverse transformer being coupled to the inverse quantizer and performing an inverse transformation of the plurality of transformed vectors to form a plurality of vectors;

d) a vector unformatter being coupled to the inverse transformer and generating a plurality of quantized transform coefficients from the plurality of vectors; and

e) a second inverse quantizer being coupled to the vector unformatter and generating the plurality of transform coefficients from the plurality of quantized transform coefficients using a second quantization level, wherein the first quantization level is greater than the second quantization level.

17. The apparatus according to claim 14, wherein the converter comprises an inverse extended quantizer, said inverse extended quantizer comprising:

a) a vector quantization indices reorderer being coupled to the decoder and performing a vector quantization indices reordering of the plurality of samples to form a plurality of quantized vectors;

b) an inverse vector quantization being coupled to the vector quantization indices reorder and performing an inverse vector quantization of the plurality of quantized vectors to form a plurality of limited dimension vectors;

c) a dimension normalizer being coupled to the

5

10

15

20

25

30

35

40

45

50

55

60

65

70

75

80

85

90

95

100

105

110

115

120

125

130

135

140

145

150

155

160

165

170

175

180

185

190

195

200

205

210

215

220

225

230

235

240

245

250

255

260

265

270

275

280

285

290

295

300

305

310

315

320

325

330

335

340

345

350

355

360

365

370

375

380

385

390

395

400

405

410

415

420

425

430

435

440

445

450

455

460

465

470

475

480

485

490

495

500

505

510

515

520

525

530

535

540

545

550

555

560

565

570

575

580

585

590

595

600

605

610

615

620

625

630

635

640

645

650

655

660

665

670

675

680

685

690

695

700

705

710

715

720

725

730

735

740

745

750

755

760

765

770

775

780

785

790

795

800

805

810

815

820

825

830

835

840

845

850

855

860

865

870

875

880

885

890

895

900

905

910

915

920

925

930

935

940

945

950

955

960

965

970

975

980

985

990

995

1000

1005

1010

1015

1020

1025

1030

1035

1040

1045

1050

1055

1060

1065

1070

1075

1080

1085

1090

1095

1100

1105

1110

1115

1120

1125

1130

1135

1140

1145

1150

1155

1160

1165

1170

1175

1180

1185

1190

1195

1200

1205

1210

1215

1220

1225

1230

1235

1240

1245

1250

1255

1260

1265

1270

1275

1280

1285

1290

1295

1300

1305

1310

1315

1320

1325

1330

1335

1340

1345

1350

1355

1360

1365

1370

1375

1380

1385

1390

1395

1400

1405

1410

1415

1420

1425

1430

1435

1440

1445

1450

1455

1460

1465

1470

1475

1480

1485

1490

1495

1500

1505

1510

1515

1520

1525

1530

1535

1540

1545

1550

1555

1560

1565

1570

1575

1580

1585

1590

1595

1600

1605

1610

1615

1620

1625

1630

1635

1640

1645

1650

1655

1660

1665

1670

1675

1680

1685

1690

1695

1700

1705

1710

1715

1720

1725

1730

1735

1740

1745

1750

1755

1760

1765

1770

1775

1780

1785

1790

1795

1800

1805

1810

1815

1820

1825

1830

1835

1840

1845

1850

1855

1860

1865

1870

1875

1880

1885

1890

1895

1900

1905

1910

1915

1920

1925

1930

1935

1940

1945

1950

1955

1960

1965

1970

1975

1980

1985

1990

1995

2000

2005

2010

2015

2020

2025

2030

2035

2040

2045

2050

2055

2060

2065

2070

2075

2080

2085

2090

2095

2100

2105

2110

2115

2120

2125

2130

2135

2140

2145

2150

2155

2160

2165

2170

2175

2180

2185

2190

2195

2200

2205

2210

2215

2220

2225

2230

2235

2240

2245

2250

2255

2260

2265

2270

2275

2280

2285

2290

2295

2300

2305

2310

2315

2320

2325

2330

2335

2340

2345

2350

2355

2360

2365

2370

2375

2380

2385

2390

2395

2400

2405

2410

2415

2420

2425

2430

2435

2440

2445

2450

2455

2460

2465

2470

2475

2480

2485

2490

2495

2500

2505

2510

2515

2520

2525

2530

2535

2540

2545

2550

2555

2560

2565

2570

2575

2580

2585

2590

2595

2600

2605

2610

2615

2620

2625

2630

2635

2640

2645

2650

2655

2660

2665

2670

2675

2680

2685

2690

2695

2700

2705

2710

2715

2720

2725

2730

2735

2740

2745

2750

2755

2760

2765

2770

2775

2780

2785

2790

2795

2800

2805

2810

2815

2820

2825

2830

2835

2840

2845

2850

2855

2860

2865

2870

2875

2880

2885

2890

2895

2900

2905

2910

2915

2920

2925

2930

2935

2940

2945

2950

2955

2960

2965

2970

2975

2980

2985

2990

2995

3000

3005

3010

3015

3020

3025

3030

3035

3040

3045

3050

3055

3060

3065

3070

3075

3080

3085

3090

3095

3100

3105

3110

3115

3120

3125

3130

3135

3140

3145

3150

3155

3160

3165

3170

3175

3180

3185

3190

3195

3200

3205

3210

3215

3220

3225

3230

3235

3240

3245

3250

3255

3260

3265

3270

3275

3280

3285

3290

3295

3300

3305

3310

3315

3320

3325

3330

3335

3340

3345

3350

3355

3360

3365

3370

3375

3380

3385

3390

3395

3400

3405

3410

3415

3420

3425

3430

3435

3440

3445

3450

3455

3460

3465

3470

3475

3480

3485

3490

3495

3500

3505

3510

3515

3520

3525

3530

3535

3540

3545

3550

3555

3560

3565

3570

3575

3580

3585

3590

3595

3600

3605

3610

3615

3620

3625

3630

3635

3640

3645

3650

3655

3660

3665

3670

3675

3680

3685

3690

3695

3700

3705

3710

3715

3720

3725

3730

3735

3740

3745

3750

3755

3760

3765

3770

3775

3780

3785

3790

3795

3800

3805

3810

3815

3820

3825

3830

3835

3840

3845

3850

3855

3860

3865

3870

3875

3880

3885

3890

3895

3900

3905

3910

3915

3920

3925

3930

3935

3940

3945

3950

3955

3960

3965

3970

3975

3980

3985

3990

3995

4000

4005

4010

4015

4020

4025

4030

4035

4040

4045

4050

4055

4060

4065

4070

4075

4080

4085

4090

4095

4100

4105

4110

4115

4120

4125

4130

4135

4140

4145

4150

4155

4160

4165

4170

4175

4180

4185

4190

4195

4200

4205

4210

4215

4220

4225

4230

4235

4240

4245

4250

4255

4260

4265

4270

4275

4280

4285

4290

4295

4300

4305

4310

4315

4320

4325

4330

4335

4340

4345

4350

4355

4360

4365

4370

4375

4380

4385

4390

4395

4400

4405

4410

4415

4420

4425

4430

4435

4440

4445

4450

4455

4460

4465

4470

4475

4480

4485

4490

4495

4500

4505

4510

4515

4520

4525

4530

4535

4540

4545

4550

4555

4560

4565

4570

4575

4580

4585

4590

4595

4600

4605

4610

4615

4620

4625

4630

4635

4640

4645

4650

4655

4660

4665

4670

4675

4680

4685

4690

4695

4700

4705

4710

4715

4720

4725

4730

4735

4740

4745

4750

4755

4760

4765

4770

4775

4780

4785

4790

4795

4800

4805

4810

4815

4820

4825

4830

4835

4840

4845

4850

4855

4860

4865

4870

4875

4880

4885

4890

4895

4900

4905

4910

4915

4920

4925

4930

4935

4940

4945

4950

4955

4960

4965

4970

4975

4980

4985

4990

4995

5000

5005

5010

5015

5020

5025

5030

5035

5040

5045

5050

5055

5060

5065

5070

5075

5080

5085

5090

5095

5100

5105

5110

5115

5120

5125

5130

5135

5140

5145

5150

5155

5160

5165

5170

5175

5180

5185

5190

5195

5200

5205

5210

5215

5220

5225

5230

5235

5240

5245

5250

5255

5260

5265

5270

5275

5280

5285

5290

5295

5300

5305

5310

5315

5320

5325

5330

5335

5340

5345

5350

5355

5360

5365

5370

5375

5380

5385

5390

5395

5400

5405

5410

5415

5420

5425

5430

5435

5440

5445

5450

5455

5460

5465

5470

5475

5480

5485

5490

5495

5500

5505

5510

5515

5520

5525

5530

5535

5540

5545

5550

5555

5560

5565

5570

5575

5580

5585

5590

5595

5600

5605

5610

5615

5620

5625

5630

5635

5640

5645

5650

5655

5660

5665

5670

5675

5680

5685

5690

5695

5700

5705

5710

5715

5720

5725

5730

5735

5740

5745

5750

5755

5760

5765

5770

5775

5780

5785

5790

5795

5800

5805

5810

5815

5820

5825

5830

5835

5840

5845

5850

5855

5860

5865

5870

5875

5880

5885

5890

5895

5900

5905

5910

5915

5920

5925

5930

5935

5940

5945

5950

5955

5960

5965

5970

5975

5980

5985

5990

5995

6000

6005

6010

6015

6020

6025

6030

6035

6040

6045

6050

6055

6060

6065

6070

6075

6080

6085

6090

6095

6100

6105

6110

6115</p

21

EP 0 833 517 A2

22

gions to form the image.

5

10

15

20

25

30

35

40

45

50

55

12

FIG. 1

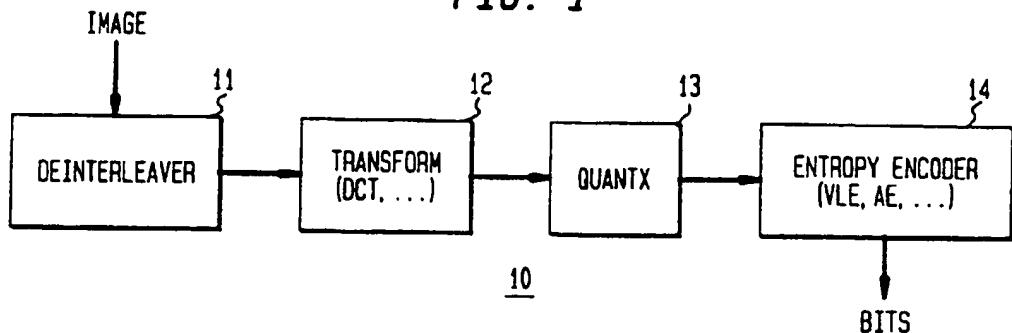


FIG. 2

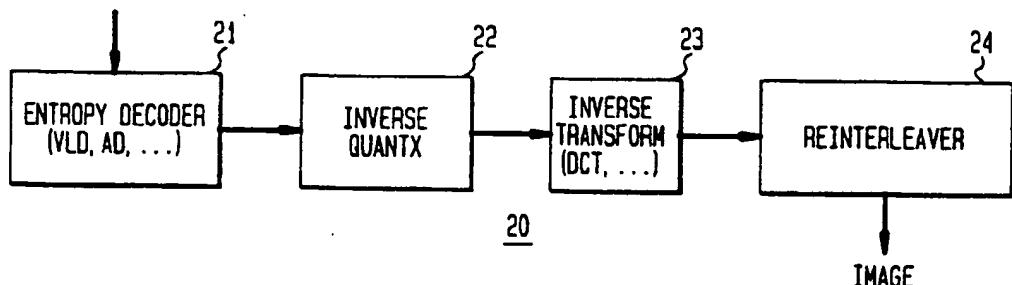


FIG. 3

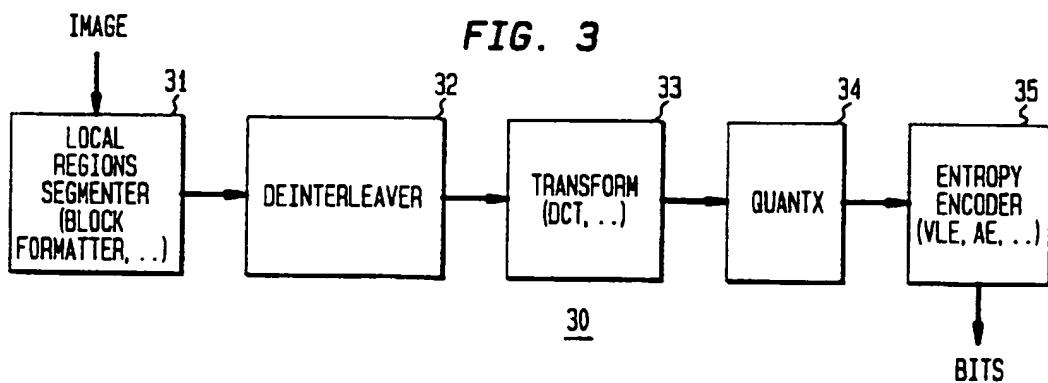


FIG. 4

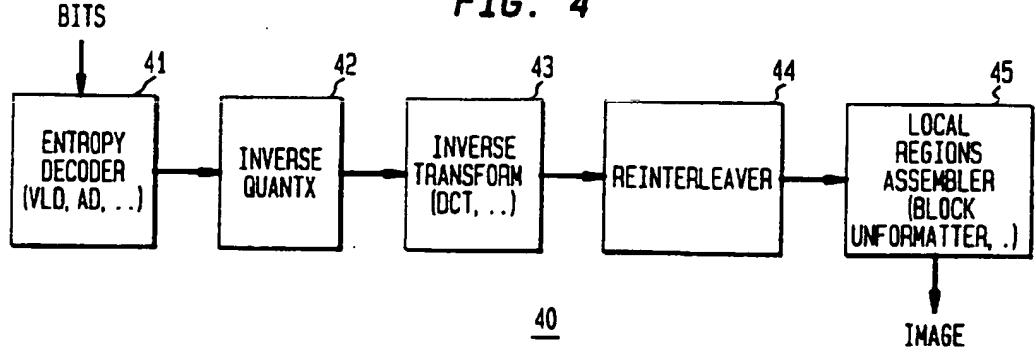


FIG. 5

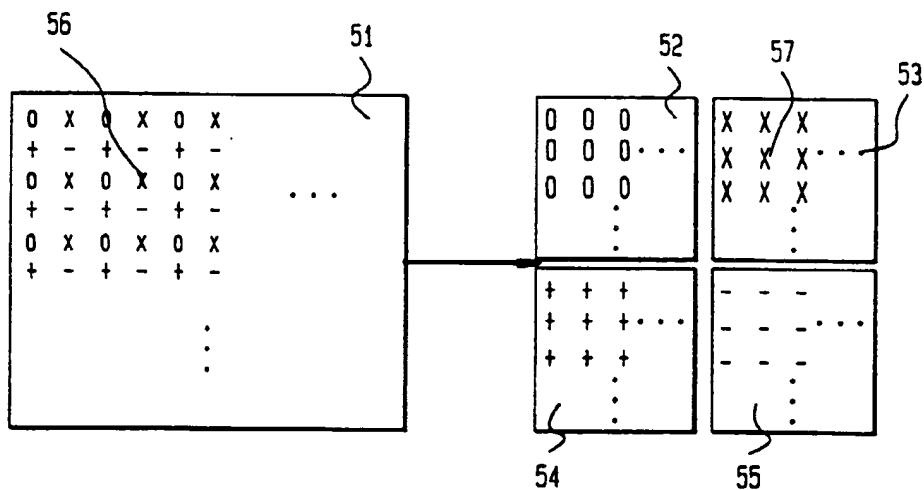


FIG. 6

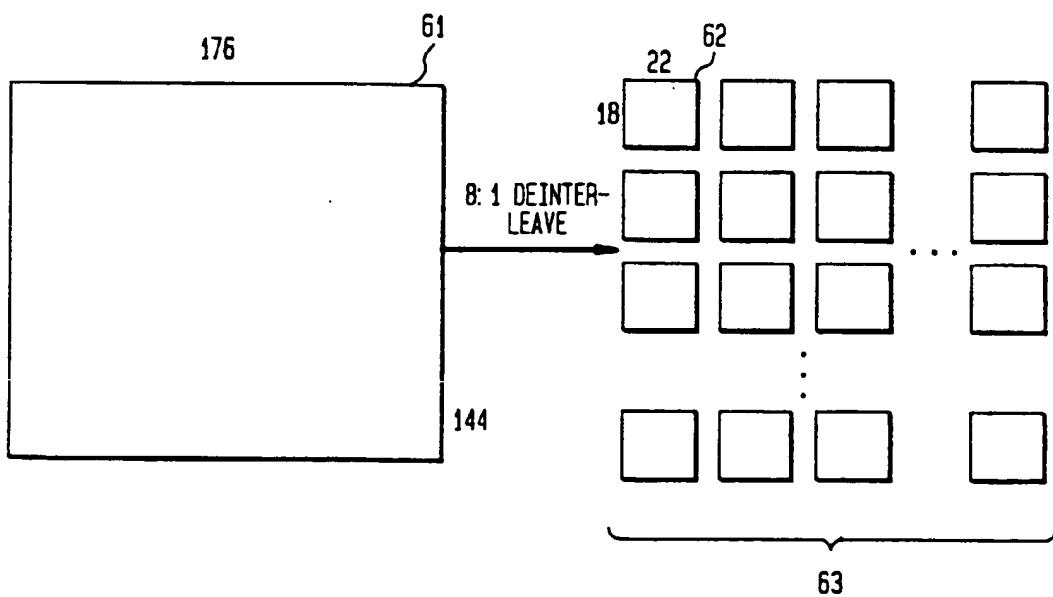


FIG. 7

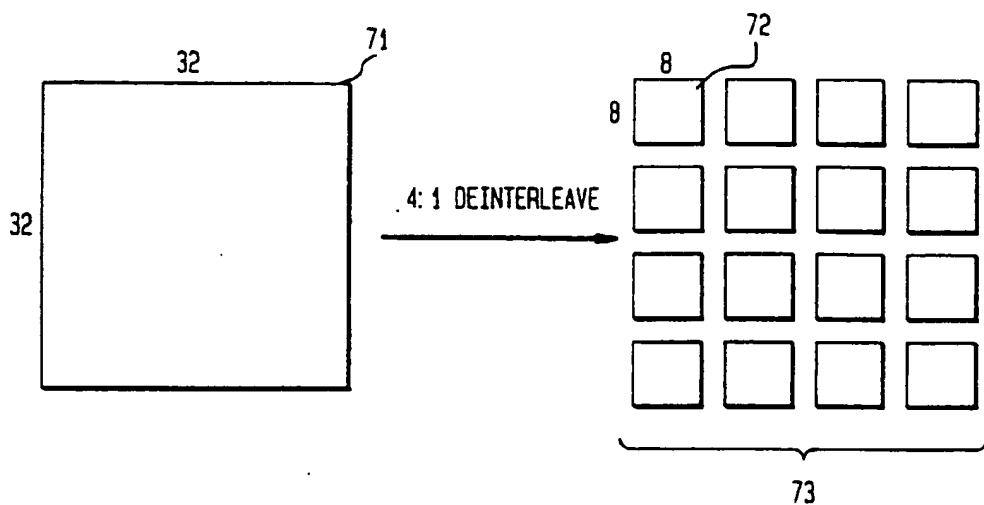
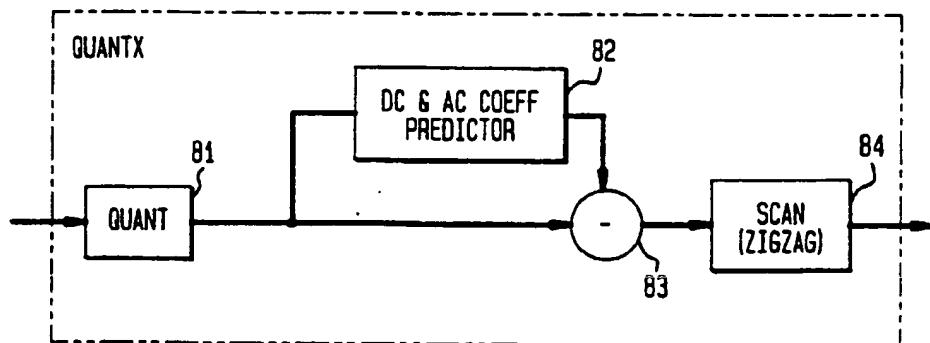
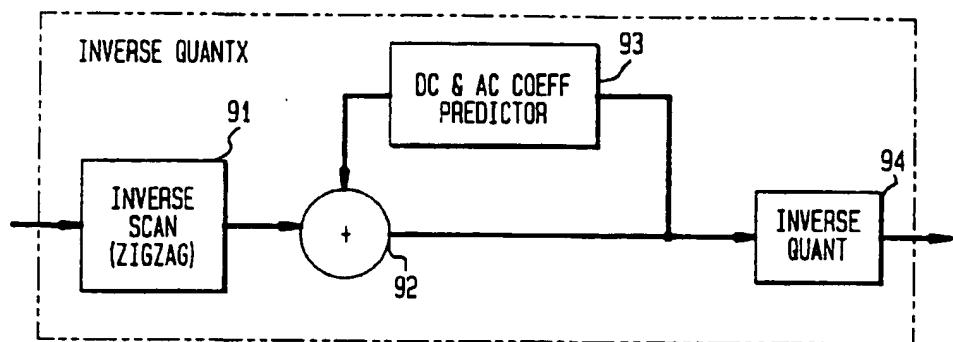


FIG. 8



80

FIG. 9



90

FIG. 10

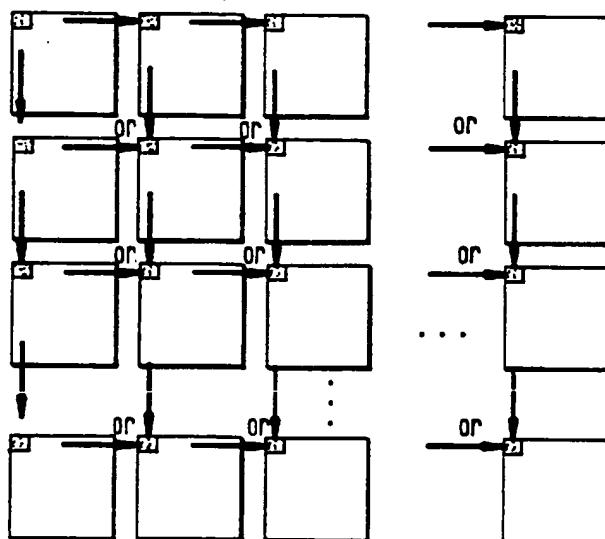


FIG. 11

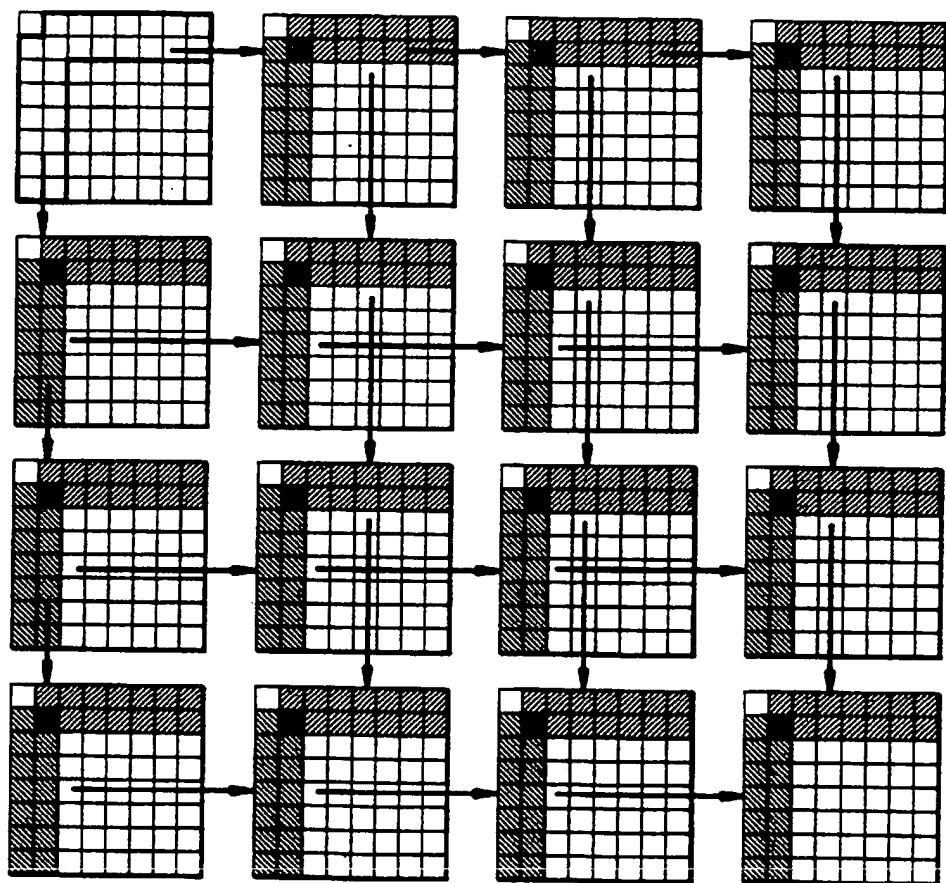


FIG. 12

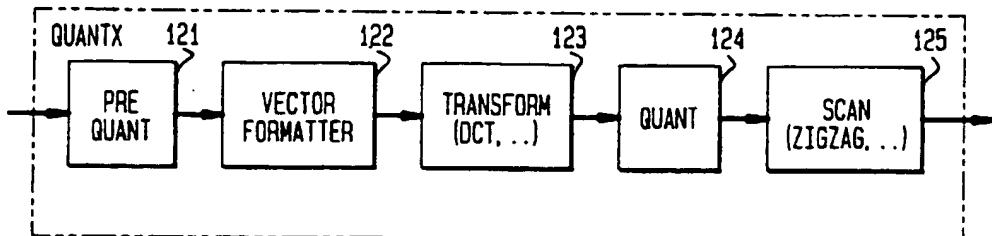
120

FIG. 13

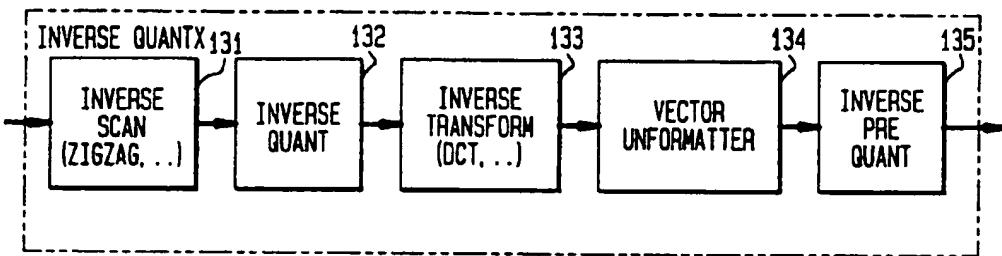
130

FIG. 14

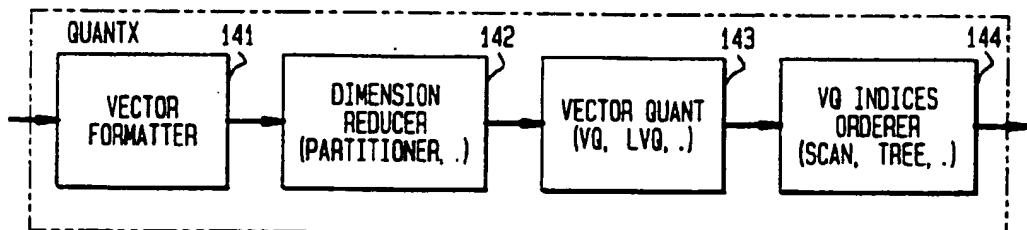
140

FIG. 15

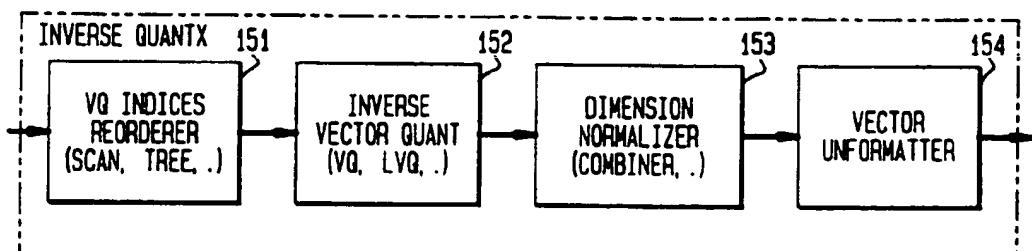
150

FIG. 16

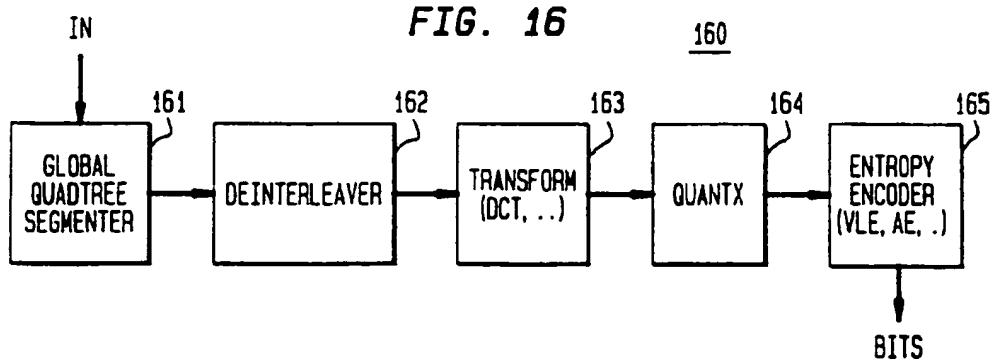


FIG. 17

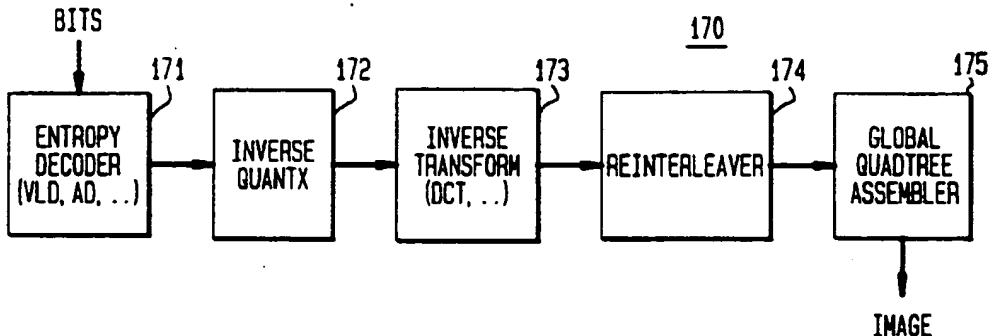


FIG. 18

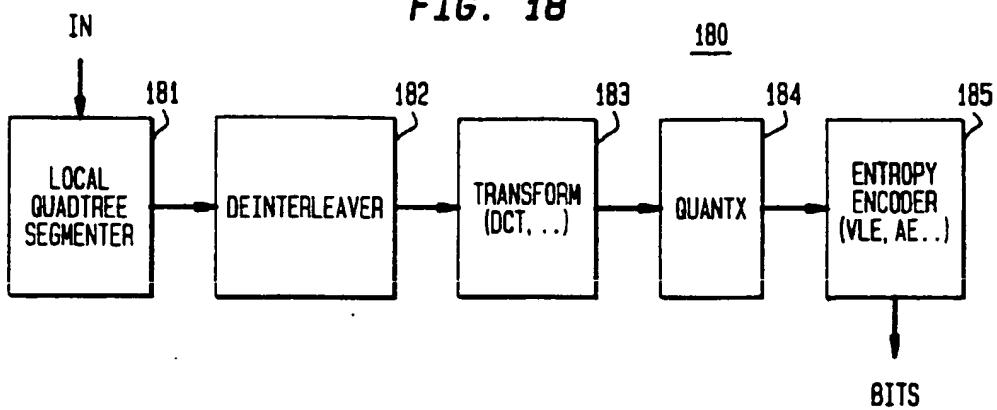


FIG. 19

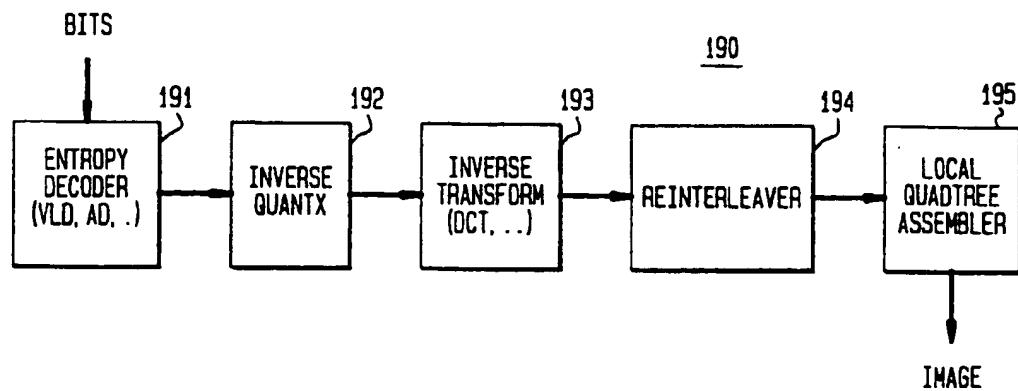


FIG. 20

